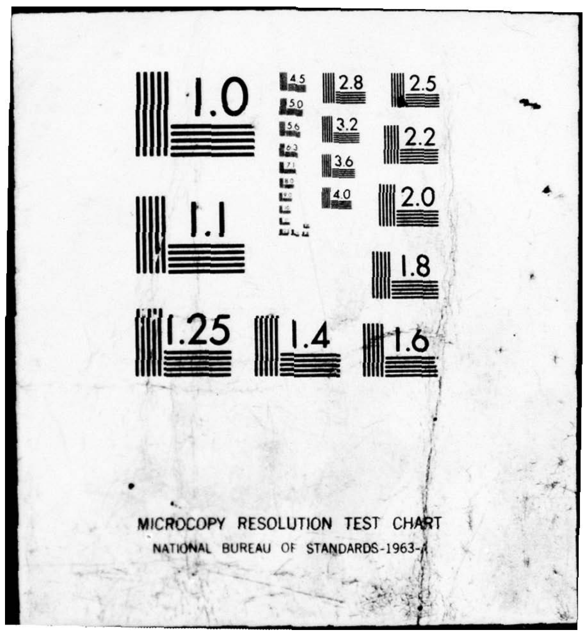
AD-A044 939 ROME AIR DEVELOPMENT CENTER GRIFFISS AFB N Y F/G 9/2
HANDBOOK FOR CONVERSION OF GCOS III APPLICATION PROGRAMS TO THE--ETC(U)
SEP 77 J KALYNYCZ

UNCLASSIFIED RADC-TR-77-241 NL

1 OF 1
AD
A044939

END
DATE
FILMED
11-77
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12
B.S.

RADC-TR-77-241
In-House Report
September 1977

HANDBOOK FOR CONVERSION OF GCOS III APPLICATION PROGRAMS TO THE GCOS ENVIRONMENT SIMULATOR OF MULTICS

John Kalynycz, Jr.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the Office of Information (OI), RADC, and approved for release to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED:

SAMUEL L. RUPLE, Captain, USAF
Chief, R&D Computer Facility
Information Sciences Division

APPROVED:

ROBERT D. KRUTZ, Colonel, USAF
Chief, Information Sciences Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy.
Retain or destroy.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 14 RADC-TR-77-241

2. GOVT ACCESSION NO.:

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): 6 HANDBOOK FOR CONVERSION OF GCOS III APPLICATION PROGRAMS TO THE GCOS ENVIRONMENT SIMULATOR OF MULTICS

5. TYPE OF REPORT & PERIOD COVERED: In-House Report

6. PERFORMING ORG. REPORT NUMBER: N/A

7. AUTHOR(s): 10 John/Kalynycz, Jr

8. CONTRACT OR GRANT NUMBER(s): N/A 17 42

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Rome Air Development Center (ISFO)
Griffiss Air Force Base, New York 13441

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 16
Program Element 62702F
~~Job Order No.~~ 55810218

11. CONTROLLING OFFICE NAME AND ADDRESS: 11
Rome Air Development Center (ISFO)
Griffiss Air Force Base, New York 13441

12. REPORT DATE: Sep[redacted]77

13. NUMBER OF PAGES: 28

14. MONITORING AGENCY NAME & ~~ADDRESS~~ (if different from Controlling Office): Same 12 34p.

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Program Transfer
Data File Transfer
Utility Services

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This handbook, for the conversion of application programs, presents information derived from a comprehensive analysis of differences in capabilities and procedures between the GCOS and the GCOS Environment Simulator. It is intended to assist applications programmers at RADC in the transfer and/or conversion of applications programs from GCOS to the GCOS Environment Simulator of MULTICS.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

309 050

LB

TABLE OF CONTENTS

SECTION 1
INTRODUCTION

This conversion handbook is intended to aid programmers involved in converting application programs for the Honeywell General Comprehensive Operating Supervisor to the GCOS Environment Simulator of Multics. To facilitate optimum usage of this handbook, it has been intentionally attempted to restrict the handbook to those applications software aspects which apply to the RADC configurations and GCOS Release G. Information that applies only to later GCOS System releases has not been specifically addressed.

It should be clearly understood that this handbook is not intended to be an explanation of either GCOS or Multics. It is basically a comparison of the features of the two systems. In this regard, the comparisons are restricted to those which are, or might be, of particular interest to an applications programmer involved in the conversion process.

The guidance supplied herein is directed primarily at explaining the features, limitations, and use of all pertinent applications aspects of the GCOS. However, it has been assumed that the primary user of this handbook will be an applications programmer who is more knowledgeable of the GCOS than the Multics. Consequently, there has been an obvious effort to present the information in such a way that it can easily be related to a programmer's present knowledge of GCOS. This is achieved by emphasizing GCOS Environment Simulator characteristics while, at the same time, maintaining a consistent association with the programmer's GCOS knowledge by illustrating how the GCOS Environment Simulator characteristics relate to the GCOS characteristics.

The technical information is contained in Sections 2 through 5 and is supplemented by the examples and other pertinent information of Appendixes A and B. This information was derived from documentation and computer run comparative analysis studies of the respective operating systems; FORTRAN, COBOL, and JOVIAL compilers; system sort; system utilities; the loaders; physical transfer of data to storage media; and available conversion aids. Most of the computer run activities for these analyses were performed under Series 6000 Software Release G.

The remaining paragraphs provide a brief overview of other portions of the handbook.

Section 2 is a comparison of the respective systems in those areas which are of interest to an applications programmer. Introductory portions discuss both hardware and software areas of difference, including such critical topics as design philosophy, organization, and operation of GCOS and Multics. Next, this

section provides a comparison of operating system services where Master Mode Entry(MME) are compared to their closest counterpart in the GCOS Environment Simulator. This is followed by a very important comparison of GCOS control cards with control cards available with the GCOS Environment Simulator and the options and parameters that apply to each card; minimum control card requirements for a local batch job and/or a Multics process.

Section 3 serves to correlate the control card functions available under native GCOS to comparable functions using the GCOS Environment Simulator; to note important differences where they exist; and to note those functions which are unique to GCOS.

Section 4 briefly describes the GCOS daemon which provides batch processing and some job scheduling facilities.

Section 5 discusses general I/O and file handling considerations of GCOS and MULTICS; compares the capabilities of magnetic tape disk, and unit record device for the two systems; and discusses the conversion aids which are available for use with magnetic tapes and cards. This information is intended to sufficiently identify and describe those properties of the two systems which must be known for any data file conversion effort.

Two appendixes are provided to supplement the information provided in sections 1 through 5. Appendix A lists all referenced documents. Appendix B contains examples to illustrate the required JCL when using the GCOS Environment Simulator.

SECTION 2
OPERATING SYSTEM

2.1 Introduction. The purpose of this introduction is to provide the reader with some insight into the basic differences between GCOS and the GCOS Environment Simulator. The emphasis is not on the features which are available in either of the systems, but, rather on those areas of difference which impact the transfer of programs from one system to another. The *H6000* General Comprehensive Operating Supervisor(GCOS) supports multiprogramming, local and remote batch job entry and a time-sharing system. Multics is a time-sharing system primarily oriented toward interactive users wherein the GCOS Environment Simulator is one of several Multics facilities that work together to provide the ability for some GCOS jobs to run under the control of Multics. Other facilities in this group include the GCOS Daemon and several commands that can be used for manipulating GCOS files. The Simulator runs as an *unprivileged* user program under Multics and depends on Multics facilities for all privileged operations such as I/O. GCOS features not simulated include:

GCOS Time Sharing

Transaction Processing

I-D-S

Remote I/O(except to the user's terminal)

Functions of the GCOS file system that have no counterpart in the Multics file system.

Multics is a time sharing system, primarily oriented toward interactive users, wherein the user may invoke the GCOS Environment Simulator(via the GCOS Command) to run a single GCOS *job, immediately*, in the user's process. However, while the job is running, the user's process is completely dedicated to it and is not able to execute other Multics commands.

The *Multics batch* processing facility allows a user to queue an absentee job to execute at some later time. This process assumes no interaction or that the user has anticipated this and placed appropriate responses in the absentee input(absin) file. Output is directed to an absentee(absout) file to be printed some time later by the user.

In addition to the interactive and absentee processes, there exists another process called the daemon process. This process is controlled by the operations staff, logged in/out by a Multics operator. These processes perform functions that are not allowed in a user's process for reasons of security/efficiency. User processes normally are not allowed to attach the on-line

printers, card reader and card punch. User processes queue requests for the input/output of files which the daemon carries out in the order that they appear in the queues. Briefly, the GCOS daemon aids the simulator of a GCOS environment by allowing standard GCOS jobs to be submitted from either punched cards or IMCV magnetic tapes in GCOS standard system format.

2.1.1 Hardware Features. Few, if any of the significant differences between GCOS and the GCOS Environment Simulator software originate in differences in hardware design. Presently, the R&D Computer Facility maintains two computer operating systems, GCOS and Multics on two similar but separate hardware configurations(H-6180) which share peripherals such as tape drives, card readers, card punch and high-speed printers. Paragraphs 2.1.1.1 through 2.1.1.3 provide a comparison of 3 areas of hardware design differences(or similarities) which lessen the impact of incompatibility between the two systems.

2.1.1.1 Multiple Processors. One advantage of H6000 systems is the availability of up to four processors maximum for any model of the H6000. Likewise, all versions of the GCOS 6000 Operating System and Multics support multiprocessing.

Multiple processors can provide increased throughput for compute bound systems and increase system reliability by allowing operations to continue when one processor fails, where a single processor system is "dead" until its processor can be repaired. It has been proposed that RADC need not maintain two computer operating systems: GCOS and Multics. Perhaps all programs could be run on a dual processor Multics by fully utilizing the GCOS Environment Simulator thus reducing the hardware configurations necessary to maintain two systems and cutting operating costs.

2.1.1.2 Relocation Registers. Another advantage of H6000 systems is the availability of a hardware relocation register or "base address register" (BAR). This feature allows reasonably sophisticated scheduling techniques in an operating system when it becomes necessary to preempt jobs, to remove them from primary storage(core) to secondary storage(drum or disk) and to later restore them to primary memory and resume their operation.

With most computer systems, the limiting physical resource is main memory. The amount of main memory available is a major factor in determining the performance of a system. The problems associated with "swapping" large files in and out of main memory severly limit system performance. Even when files are not all large, there still remains the difficult problem of core management. Since Multics allows users to create and/or manipulate large segments, it is neither feasible nor desirable to have an entire segment in main memory when in use. Only those segment "pages" (1024 words) currently needed are in memory at any one time. The Multics software moves data rapidly into and out of main memory as required, completely automatically, so that

the user has the illusion of a memory much larger than physical memory.

2.1.1.3 Storage Protection. Some of the most significant differences between GCOS and Multics stem from the different approaches chosen by the respective hardware designers in implementing a storage protection feature for primary storage (core).

The H6000 GCOS system uses its hardware BAR to provide storage protection for slave programs. The BAR is divided into two parts; the most significant half specifies a starting address(a multiple of 1024 words) while the least significant half contains a count of the number of 1024 word blocks allocated. With the processor in slave mode, the hardware automatically compares the address of every memory access to the contents of the BAR. If the address is within the specified limits, the operation is allowed to continue normally. However, if it is not within the specified limits, a machine fault is generated and the memory access is denied, thus providing both read and write protection.

The Multics operating system employs a ring protection mechanism similar to the two-level(master/slave) protection feature of GCOS to control its users and to protect itself. What is unique about the Multics system is that there are eight levels of protection (ring 0-7) rather than two; ring 0 being the most privileged, and ring 7 the least privileged. The operating system resides in the most privileged rings(0 through 2) while users generally execute in the lesser privileged rings(3 through 7). All segments in the system have ring protection attributes associated with them. A set of three ring levels called ring brackets in addition to the access control list allow a separate ring level to be specified for writing, reading and/or calling and executing a segment. Thus, exact and precise control of every piece of information in the system is enforced by the hardware on every reference to the information.

2.2 System Software Comparison. The most significant difference between GCOS and Multics is in the philosophy regarding the use of frequently required routines. The majority of segments on the Multics operating system(i.e., the operating supervisor, compilers, library routines) are pure procedure. Only one copy of a pure procedure segment is needed in main memory, no matter how many users are executing it. With only one copy required, the amount of data transfer is greatly reduced and main memory is more efficiently utilized. Thus the cost performance factor of the system is significantly enhanced.

GCOS uses reentrant code only in selected areas of the master mode operating system. The slave service area(SSA) of GCOS is based on the belief that multiple copies of some of the

master mode user service modules are more efficient than the same module would be if it were made reentrant and shared by users concurrently. In slave mode, GCOS is forced to use the multiple copy approach because of the hardware requirement that the core allocation for each program be a contiguous area.

2.2.1 Supervisor Services. GCOS provides supervisor services in the master mode entry(MME) form for users to call on in the execution of their programs. Not all native GCOS MME's are implemented for the GCOS Environment Simulator; others are only partially implemented. Section IV of General Comprehensive Operating Supervisor (GCOS) order no. DD19 Rev. Ø describes the MME routines as implemented under native GCOS. This material is repeated in Section IV of GCOS Environment Simulator order no. ANØ5A Rev. Ø followed by brief statements regarding exceptions to the native GCOS implementation.

The following MME routines are fully implemented:

GEENDC
GEFINI
GEMREL
GERELS
GERETS

If the following MME routines are requested, no action is taken other than to return to the slave program:

GECHEK
GELOOP
GEPRIO
GERELC
GEROAD
GESNAP
GEWAKE

The following MME routines are not implemented and if called for will abort the slave program:

GEFILS
GEFRCE
GEIDSE
GELBAR
GENEWS
GEROLL
GESNUM
GESPEC

The following MME routines abort the activity if any errors occur as error codes and error returns are not implemented:

GECALL
GERSTR
GESAVE.

The following MME's are implemented with exceptions as noted:

GEBORT - Request to abort program.

In native GCOS, if all I/O activity is not completed prior to executing this MME, the I/O may be lost. In the GCOS Environment Simulator, no I/O is lost since all I/O is synchronous.

GEFADD - Physical File Address Request
GEFCON - File Control Block Request

All devices(except magnetic tapes), which are handled by Multics I/O, are simulated in a Multics segment. Physical device addresses (in the GCOS sense) have no meaning in the Simulator. The device address is dummied in both MME routines.

GEFSYE - File System Entry

The GCOS file management System is not used or supported. All files and catalogs are retained as Multics segments in the Multics File Management System. The MME GEFSYE performs equivalent functions in the Multics File Management System for the following functions:

Catalog Create
File Create
File Purge
Catalog Modify
File Modify
File Release
File Query.

All other functions abort the slave job.

GEINFO - Information GCOS Entry

The buffer function is not supported and causes a slave job abort. The List Pointer Word(LPW) function is implemented. Information returned is meaningful only for SYSOUT lines and Time-of-day(TOD) of activity start.

GEINOS - Input/Output Request

Multics(consequently, GCOS Environment Simulator) tape buffer size is limited to 1632 words. Interslave Communication(INTERCOM) is not supported as only one slave job is allowed.

GELAPS - Elapsed Time Request

The MME GELAPS routine returns the amount of Multics processor time that has been expended.

GEMORE - Request for Additional Peripherals or Memory

The option to request a cataloged file results in the creation of a Multics segment with Multics permissions.

GEROUT - Remote Output Record

The operation codes implemented are as follows:

Ø3 - Direct Access Output
Ø4 - Direct Access Output, Then Input
Ø5 - User Program Inquiry to Terminal
Ø6 - Program Request Terminal Type
17 - Program Request Line Disconnect
2Ø - Direct Access Current Line Status.

The options not implemented fall into the areas of paper tape manipulation or line switching.

GESYOT - Write on Sysout

Remote and backdoor Sysout are not permitted. The format of the execution report was changed to reflect Multics information.

GEUSER - User-supplied MME

Each interactive user can provide an individual handler for this MME. Guidelines for writing a module and some format restrictions of a module may be found in the listings of procedures named gcos_mme_xxxx(where xxxx is the name of any supported MME). Using the Multics search rules, a search is made For a Multics object segment named mme_geuser.

.EMM - Enter Master Mode

The GCOS Environment Simulator does not support privity. This MME always aborts the slave program.

SECTION 3
CONTROL CARDS

3.1 The GCOS control cards provide the operating system with the information required to execute the job of which they are a part. The term "job" has identical meaning for both systems. Some of the common features of job control cards are: the assignment of job and activity identification(where activity is a subdivision of a job); designation of the hardware resources required for execution and their manner of use; user identification; assignment of level of job priority; and the statement of performance limits or estimates of the maximum quantities of system resources to be allowed.

The implementation of control cards for GCOS involves eight categories consisting of more than 100 cards which are available for execution of activities of a job. Section V of GCOS ENVIRONMENT SIMULATOR (ANØ5A) describes in detail all control cards available with the GCOS Environment Simulator and the options and parameters that apply to each card. It has been estimated that 82 percent of the 1Ø7 control cards available in native GCOS are also available via the GCOS Environment Simulator.

Each input deck requires three control cards that serve to delimit the deck as well as perform some accounting functions. These cards are SNUMB, IDENT and ENDJOB. The SNUMB card carries installation-sensitive parameters and, therefore, is supplied by operations personnel at the time the job is submitted for processing for local batch in native GCOS and via the Multics GCOS Daemon. If the user submits a GCOS job via the interactive Multics process, the SNUMB is required and is used by the Simulator to identify the job internally. In native GCOS at least one IDENT control card must immediately follow the SNUMB control card. This card remains effective until cancelled through subsequent encountering of another IDENT card in the job stream. While GCOS allows both accounting and file access information to be changed within a job(via multiple IDENT and USERID control cards), Multics does not permit such changes within a process. Only one IDENT card is allowed in a job stream run in the Simulator and the GCOS Userid card(required if access is made to permanent files) is accepted, but ignored, by the Simulator. The ENDJOB card explicitly indicates the end of a job in both systems; the ***EOF is ignored by the Simulator unless the job is submitted via the GCOS daemon whereas in the absence of an ENDJOB card, the ***eof card is recognized as a job separator.

The following paragraphs discuss those control cards which are not available for use in the GCOS Environment Simulator. However, the user is further advised to pay particular attention to the control card descriptions(Section V of ANØ5A Rev. Ø) with respect to unique restrictions/limitations.

The DUMMY control card of the Basic Input deck category is not supported as MME GENEWS is not implemented. The GCOS Environment Simulator is restricted to running one job per Multics process and does not allow a user program to spawn independent programs of which this control card is a necessary element. No provision is made for jobs that require special privileges, such as use of the MME .EMM, therefore, the PRIVITY control card is not allowed.

The File Control Cards not supported include FORM, REPORT, REPTL, REPTR, NTAPE, PARAM, PPTP, PPTR, READ, REMOTE, and 180PK. File Control Cards are supplied by the user at execution time to specify the actual device type desired for each file as well as define file information for a program to be processed. The Environment works through the Multics input/output(and peripheral allocation) system and the Multics file structure. Job output created by activities go to SYSOUT(sysprint and syspunch) collection files, and are queued for output by the Multics I/O daemon. Consequently, control cards normally used to change forms for printer or punch output are not supported. The peripheral allocator does not permit card reader, paper tape reader or punch and removable disk packs, therefore, READ, PPTP, PPTR, and 180PK control cards are not supported. Jobs submitted to the Simulator describing such hardware features not supported will abort with "JOB DELETED NO DEVICE $ CARD xxx". The NTAPE and PARAM control cards are not implemented. The Simulator does not support remote I/O, therefore, the REMOTE control card is not supported. Finally, the TAKEc option on DATA is not supported and will result in a "WARNING: AN UNIMPLEMENTED OPTION ON $DATA CARD HAS BEEN RECOGNIZED". The DATA control card implementation is an example of a unique option not being supported and the user is referred to detailed differences in GCOS Environment Simulator and native GCOS as noted in the section describing control cards.

With the exception of two system call cards (FILSYS and IDS), all remaining categories of control cards are implemented. Neither the GCOS File System nor the Integrated Data Store are simulated by the GCOS Environment Simulator.

3.2 GCOS Local Batch Job. This paragraph illustrates the minimum required input to assemble and execute a simple GMAP program in the local batch mode of GCOS. The SNUMB card is normally supplied by computer operations personnel; it provides the identification by which the operating system tracks a job. The IDENT card provides user identification for accounting purposes. The GMAP system call control card causes the GMAP assembler to be loaded and executed using the following two GMAP statements as data. The EXECUTE card causes execution of the assembled code and would be omitted if execution were not desired. The ENDJOB card delineates the end of the job at execution time. The***EOF card delineates the end of job at job input time; i.e., notification to the GEIN software. For other languages, the GMAP card would be replaced and the source deck

would contain statements in the desired language. The following cards represent an actual job:

```
1       8       16
$       SNUMB   12345
$       IDENT   person.project
$       GMAP
        SYMDEF  START
        START   END
$       EXECUTE
$       ENDJOB
***EOF
```

SECTION 4
MULTICS GCOS DAEMON

4.1 The Multics GCOS Daemon is a facility designed to aid in the simulation of a GCOS Environment on Multics. The daemon allows standard GCOS jobs to be submitted from either punched cards or IMCV magnetic tapes in GCOS standard system format. The job described in paragraph 3.1 is representative of a standard GCOS job.

Jobs punched on cards can be submitted to the Simulator in two ways:

1. The job deck can be submitted to the GCOS daemon. The deck is given to operations to be read by an on-line card reader under control of the daemon and an absentee job is submitted to run the GCOS job.

2. The job deck can be submitted for input to a segment by using the Multics bulk card input facility(refer to 5.1.3 for information on this facility). The segment can be used as input to the Simulator in either an interactive process or in an absentee job submitted by an interactive process.

IMCV options available under native GCOS are also available under the GCOS Daemon. IMCV tapes need only be in GCOS standard system format as created by Bulk Media Conversion(BMC).

SECTION 5
PHYSICAL TRANSFER OF DATA

5.1 Introduction. The purpose of this handbook "Physical Transfer of Data" is concerned with those properties of each computer system which impact upon the ability of a programmer to transfer or convert existing data files from native GCOS to the GCOS Environment Simulator. Subsequent paragraphs discuss general I/O and file handling considerations of GCOS and GCOS Environment Simulator; compare the capabilities of magnetic tape, disk, and unit record devices for the two systems; and discuss the conversion aids which are available for use with magnetic tapes and cards. It is intended that this information will sufficiently identify and describe those properties of the two systems which must be known for any data file conversion effort.

5.1.1 General I/O and File Handling Considerations. Both systems permit the user to perform I/O operations at three levels. At one level, the user must employ the MME GEINOS; at another level, the I/O is performed via calls to the General File and Record Control (GFRC) integrated set of I/O subroutines; and at the level most commonly used by applications programmers, assembler calls, and higher level language statements are employed.

In allowing I/O operations at these three levels, native GCOS supports two access methods(logical and physical) and four data file organizations(sequential, random, indexed sequential, and integrated data store(IDS). I/O performed at the MME GEINOS cannot use the logical access method and operations performed by higher level language statements can only use the logical access method, while, at the GFRC level, both access methods can be used.

The GCOS Environment users operate on mass storage files through the facilities of the Multics storage system. Both permanent files (nonremovable) and temporary files are processed by GCOS programs in the normal manner. The Multics features of paging and segmentation are transparent to Simulator users. Since all GCOS files are actually Multics segments, access to the segments can be either by GCOS slave programs operating under the Simulator or by any Multics procedure.

Files to be used by a job executing under the Simulator can reside either in the Multics Storage System(in the form of paged segments) or on magnetic tape in the same logical structure as that under native GCOS. Files on magnetic tape can be processed by the same slave program operating under either the Simulator or native GCOS.

Data formats are identical to those for native GCOS. The

Simulator interfaces with the slave software at the MME level so that all Multics supported GCOS slave program content managers (such as GFRC) define the content and format of the data to be read or written. One exception to this is the limitation on the size of physical tape records (1632 words), due to a Multics constraint.

Standard GCOS access modes(sequential and random) are supported by the Simulator. Indexed sequential and IDS file organization are not. The GCOS File Management System is not used or supported but files and catalogs are retained as segments in the Multics File Management System. References to permanent files from PRMFL or SELECT control cards or from MME GEFSYE are mapped by the Simulator into references to segments in the Multics Storage System.

5.1.2 Utility. One method to move GCOS files(non-tss) to Multics for use in the GCOS Environment Simulator uses the utility function of both systems. This method is particularly suited for files containing object code and stored in either a sequential or random file. The following steps illustrate the processes involved in moving an object code file stored sequentially. Note, that in GCOS, the process can be accomplished in batch or CARDIN.

STEP 1: In GCOS, copy the file to magnetic tape.

```
$       IDENT
$       USERID
$       MSG2       1,THIS JOB USES TAPE NO. 12345 RING-IN
$       UTILITY
$       FUTIL      A1,B1,COPY/1F/
$       TAPE9      B1,X1S,,12345,,COPY-WRITE,,DEN8
$       PRMFL      A1,R,S,catalog-file string
$       ENDJOB
***EOF
```

Determine the size(in blocks) of the file in GCOS to calculate a bit count in STEP 2.

STEP 2: In Multics, prepare file space as follows:

```
r 842 ... (The computer is ready to accept a command.)
create segmentname
r 843 ...
set_bit_count segmentname count
r 844 ...
```

where count is the bit count, in decimal, calculated as follows:

count = number of blocks X 320 words/block X 36 bits/word

STEP 3: The Multics editor "qedx" is used to create a segment containing the job stream to be run utilizing the GCOS Environment Simulator to copy a file from the magnetic tape created in STEP 1 to a Multics segment. To create the segment called copy, type the following:

```
qedx
$a
```

Here "qedx" calls the editor; since this is a new segment, it is not necessary to name it until you are ready to store a permanent copy. Note that the computer will not print a response.

Then a text line can be entered followed by a carriage return. Next another text line is entered and the process continues until all the text lines(job deck segment) have been entered.

The entire job deck segment follows:

```
$       snumb     copy1
$       ident     project-id,person-id
$       utility
$       futil     a1,b1,copy/1f/
$       tape9     a1,xls,,12345,,12345,,den8 -noring
$       prmfl     b1,w,s,>udd>project-id>person-id>segmentname
$       endjob
***eof
```

A "backslash f" (\f) should be typed on the line after the last line to signify the end of input. For example,

```
***eof
```

is the last line of the text to be entered, the remainder of the terminal session would look like this:

```
***eof
\f
w copy       ("w"(write) is typed to have the computer store the
             segment with the name copy)
q            ("q"(quit) is typed to tell the computer you are
             finished editing)
r 855 ...    (The computer is ready to accept a command)
```

At this point the Multics command to invoke the GCOS Environment Simulator to run the utility job, immediately, is entered:

```
gcos copy
```

Upon completion of the process, the Multics segment can be used in a job stream generated in STEP 4.

<u>STEP 4</u>: In Multics, create whatever job stream was necessary to run the job as was done in native GCOS. Replace GCOS file strings with Multics pathnames.

5.1.3 If a GCOS card deck is to be transferred, the Multics Bulk Input/output facility may be used. This facility provides for punched card decks to be read into Multics segments. The card deck format most commonly used for data interchange between Multics and other systems(e.g.,IBM) is the Multics Card Code(MCC). Card decks to be read into Multics segments are submitted to the I/O clerk in Building 3. Each deck must begin with two keypunched control cards: an access_id_card and a deck_id card. These are used to identify the submitter to Multics and describe the deck name and format. The deck is then submitted to operations and read into System Pool Storage. The user must copy the card image segment into his directory with the copy_cards command.

Card image segments must be copied from the System Pool Storage within a reasonable time, since the segments in that area are periodically deleted.

Control Card Formats

The access_id card has the following format:

PERSON_ID.PROJECT_id

where:

PERSON_id is the registered name of the submitter. Only this person will be able to read the card image segment from the pool.

PROJECT_ID is the project name of the submitter separated from the PERSON_ID by a period and ending with a semicolon.

The deck_id card has the following format:

DECK_NAME PUNCH_FORMAT

where:

DECK_NAME is the unique name used to identify the card image segment in System Pool Storage.

PUNCH_FORMAT is the punch code conversion(MCC) to use in reading the card deck.

All characters on the control cards are mapped to lower case except those immediately following an escape character(cent sign). For example, ¢MY_¢DECK.FORTRAN will be mapped to My_Deck.fortran.

Example:

Suppose user John Doe, working on project Proj, wishes to read a FORTRAN source deck into a segment called alpha.fortran. The following is a step-by-step procedure:

1. The submitter logs in as Jdoe.Proj and verifies that alpha. fortran does not currently exist by typing:

ls -a

2. The source deck is set up with the appropiate control cards as follows:
¢JDOE. ¢PROJ;
ALPHA.FORTRAN MCC
.
.
.
source program deck in MCC format
.

.

3. The card deck is submitted to the I/O clerk.

4. The submitter then logs in as Jdoe.Proj and issues the command
copy_cards alpha.fortran
at his terminal to copy the cards into his working directory.

5. The user must do a one-for-one conversion from upper case letters to lower case and does so with the following command:

convert_characters alpha.fortran beta.fortran

6. The new segment(beta.fortran) is now ready to be edited in preparation for executing in the GCOS Environment Simulator.

For additional information about the convert_characters(cvc) command, type help cvc.

5.1.4 GCOS/Multics File Transfer Facility

This program provides a facility for transferring programs and data files (via magnetic tape) from the GCOS system to the MULTICS. Once in that environment, this software provides assistance in converting GCOS source language(e.g., FORTRAN, BASIC) to function under Multics. For more information on the use of this facility, the user may dprint the Multics on-line documentation.

```
dprint >doc>handbooks>file_transfer.runout
```

Procedures to go from GCOS to the GCOS Environment Simulator of Multics follow:

1. Under GCOS do an ascii to bcd, then a bcd to acsii to move or strip the line sequence numbers from the source file.

2. Use CONVER to write the file to tape.

```
20$:IDENT:
30$:USERID:userid$password
40$:CONVER
50$:PRMFL:IN,R,S,catalog-file string
60$:TAPE9:OT,A1D,,12345,,TRANSFER,,DEN8
70$:ENDJOB
```

3. Under Multics, notify operations of your intent to enter a transfer request. Give the following information for the tape made above:

a. Tape Number
b. Specify 7 or 9-track
c. Density(200,556,800,1600)
d. Person.Project

4. At your terminal, enter the transfer request as follows:

```
enter_transfer_request -opt- tape_no
```

where tape_no is the GCOS tape number of the reel containing the file to be copied and -opt- represents the following option:

"-f" select one or more files from a multifile tape

The files(-f) argument is followed by two integers which represent the starting file number and the number of files.

For example:
-f 1 1 indicates that file number one of one file is to be copied.

The following etr would copy one file from tape number 12345 to the working directory of Jdoe.

```
etr -f 1 1 -o Jdoe 12345
```

Invoke the list command to determine the name of the segment that is created as person_id.1

```
ls -ft 1
```

```
Jdoe.1
```

The segment thus created is in GCOS format, upper case, and access mode set to read and execute. The access mode must be changed to read and write(modify).

```
sa Jdoe.1 rw Jdoe.Proj.*
```

At this point, the segment must be reformatted and in doing so, the segment is written over.

```
reformat Jdoe.1
```

The Multics command convert_characters(cvc) to convert upper case to lower case may be used:

```
cvc lc Jdoe.1 segmentname
```

segmentname results in a Multics segment which may be edited under qedx.

APPENDIX A

REFERENCES

1. Honeywell Information Systems, Inc., Honeywell Series 60 (Level 66)/6000 General Comprehensive Operating Supervisor, DD19, Rev. 0, April 1974.

2. Honeywell Information Systems, Inc., Level 68 MULTICS GCOS Environment Simulator, AN05A, Rev. 0, December 1974.

3. Honeywell Information Systems, Inc., GCOS/MULTICS File Transfer Facility, RADC-TR-75-137, May 1975, AD A013 109.

APPENDIX B

GCOS ENVIRONMENT SIMULATOR EXAMPLES

These examples illustrate the required JCL to utilize certain GCOS features that are also available when using the GCOS Environment Simulator. Each of these examples starts a new page for ease in referencing.

```
$       snumb    dac
$       ident    5581i0218,Kalynycz
$       option   fortran
$       use      .rtyp
$       fortran  ndeck
    SOURCE

$       execute
$       dac      05
$       dac      06
$       endjob
```

Example 1 - Conversational I/O

```
$       snumb   bmdØ5
$       ident   5581iØ218,Kalynycz
$       option  fortran
$       library bd
$       use     bmdØ5r
$       entry   bmdØ5r
$       execute
$       limits  25,4Øk,,1Øk
$       tape9   bd,d1dd,,12345,bmd-r*,,den8 -noring

     DATA

$       endjob
```

Example 2 - BMD Uitilizing R* Tape

```
$       snumb   jocit
$       ident   5581iØ218,Kalynycz
$       option  jovial
$       jovial
$       select  >ml>jocit>compile

    SOURCE

$       execute
$       select  >ml>jocit>execute
$       limits  99,3Øk,,2Øk
$       endjob
```

Example 3 - JOCIT JOVIAL

```
$       snumb   cplot
$       ident   5581i0218,Kalynycz
$       option  fortran
$       forty
```

SOURCE

```
$       select  >ml>plotter>complot
$       execute
$       limits  99,30k,,10k
$       ffile   26,nstdlb,nosrls,fixlng/226,noslew,bufsiz/226
$       tape9   26,Xld,,12345,,complot,,den8 -ring
$       endjob
***eof
```

Example 4 - Complot Plotter